CLAIMS

What is Claimed is:

1. A method comprising:

stalling a call to an operating system function originating from a call module; and

determining whether said call module is in a driver area of a kernel address space of a memory.

2. The method of Claim 1 further comprising determining that said call module is not in said driver area during said determining.

3. The method of Claim 2 further comprising taking protective action to protect a computer system.

4. The method of Claim 3 further comprising providing a notification that said protective action has been taken.

5. The method of Claim 2 further comprising terminating said call.

6. The method of Claim 2 further comprising terminating a parent application comprising said call module.

7. The method of Claim 2 further comprising determining whether said call module is a known false positive.

8. The method of Claim 1 further comprising determining that said call module is in said driver area during said determining.

9. The method of Claim 1 further comprising stalling said call.

10. The method of Claim 9 further comprising:

determining that said call module is in said driver area during said determining; and

allowing said call to proceed.

11. The method of Claim 1 further comprising determining a location of said call module in said kernel address space of said memory.

12. The method of Claim 1 further comprising determining if a last mode of operation is a kernel mode.

13. The method of Claim 1 further comprising disabling loading and unloading of drivers into said kernel address space.

14. The method of Claim 13, further comprising, subsequent to said determining whether said call module is in a driver area of a kernel address space of a memory, enabling loading and unloading of said drivers into said kernel address space.

15. The method of Claim 1 wherein said driver area is static.

16. The method of Claim 1 wherein said driver area is dynamic.

17. The method of Claim 16 further comprising keeping said driver area updated as drivers are loaded and unloaded from said kernel address space.

18. A method comprising:

hooking driver load and unload functions;

obtaining loaded driver information;

determining a driver area in a kernel address space of a memory; and

determining whether a driver has been loaded into or unloaded from said kernel address space, wherein upon a determination that said driver has been loaded into or

unloaded from said kernel address space, said method further comprising updating said driver area.

19. The method of Claim 18 further comprising:

stalling a call to an operating system function originating from a call module; and

determining whether said call module is in said driver area.

20. The method of Claim 19 wherein said driver area is dynamic.

21. A computer-program product comprising a computer readable medium containing computer code comprising:

a malicious code blocking application for stalling a call to an operating system function originating from a call module; and

said malicious code blocking application further for determining whether said call module is in a driver area of a kernel address space of a memory.